Eng. & Tech. Journal, Vol.29, No.2, 2011

Different Software Components Integration using C# and Matlab Platforms

Dr.Mohammed Najm Abdullah , Dr. Hassan Awheed Jeiad* & Rabab Jassim Mohsun**

Received on: 18/2/2009
Accepted on: 2/12/2010

Abstract

The distributed computations can produce significant performance gains, yet the time and expertise needed for the low-level details of distribution is often prohibitive. Additionally, many applications rely heavily on domain-specific libraries, while it is not practicable to write an optimizing compiler each time a domain-specific library is developed. The purpose of this paper is to solve this problem by proposing a distribution system which distributes job execution over several computers using a distribution Matlab compiler, called the Matlab grid compiler.The implementation concentrates on connecting several computers known as clients to a main computer known as the administrator. A software application is built using Microsoft visual studio, then executed on a server-client network, and the results showed that integrating the components of C# and Matlab gives a valuable worth performance for the distributed computation techniques.

Keywords: Distributed Computation, Grid system, C#, Matlab.

تكامل مركبات برمجية مختلفة باستخدام لغة سي هاش وتطبيق الماتلاب

الخلاصة

ان استخدام المهام الحسابية الموزعة يمكن ان ينتج مكاسب كبيرة في الاداء, و لكن الوقت و الخبرة اللازمة للتعامل مع تفاصيل عملية التوزيع غالبا ما تكون مكلفة. بالاضافة الى ذلك فان العديد من التطبيقات تعتمد بشدة على مكتبات المجال المخصص. في حين انه ليس من العملي كتابة مترجم في كل مرة يتم فيها تعديل مكتبات المجال المخصص.الهدف من هذا البحث هو حل هذه المشكلة من خلال توفير نظام توزيع يقوم بتوزيع تنفيذ المهام عبر عدة حاسبات باستخدام مترجم خاص بالـ Matlab يسمى Matlab Grid Compiler . يتركز التنفيذ في ربط عدة حاسبات تعرف بالخوادم مع حاسبة مركزية تعرف بالمدير. تم بناء برنامج تطبيقي باستخدام ال(visual studio) وتم تنفيذه على شبكةحاسبات نوع سيد-خادم. اظهرت النتائج بان تكامل مكونات الـ C# و الـ Matlab اعطى مكاسب قيمة في اداء تقنيات المهام الموزعة.

1. Introduction.

As available computing power increases because of faster processors and faster networking, computational scientists are attempting to solve problems that were considered infeasible until recently. Grid systems are becoming more pervasive platforms for running distributed jobs to solve such problems. A grid system is an environment in which users, such as scientists, can access resources in a transparent and secure manner. In a grid system, when a user submits a job, the system runs the job on distributed resources, and enables the

* Computer Engineering & Information Technology Department, University of Technology/Baghdad
** Engineering College, University of AL- Mustansiriya /Baghdad

https://doi.org/10.30684/etj.29.2.2
2412-0758/University of Technology-Iraq, Baghdad, Iraq
This is an open access article under the CC BY 4.0 license http://creativecommons.org/licenses/by/4.0

user to access the results of the jobs when they are complete [1].

The term .Net Remoting is one of the most widely implemented technologies which represent a Microsoft Application Programming Interface (API) released with the first version of .Net Framework. An API is simply a set of functions that you can use to work with a component, application, or operating system. Typically an API consists of one or more DLLs that provide some specific functionality.

The approach of integrating MatLab with another application to make Matlab work as a distributed system has been regarded by many researches, Anne E. [2] presented the run of MATLAB processes on multiple processors, with full access to all the usual capabilities such as Toolboxes. Vikram Dham [3]. had integrated MATLAB and OPNET simulations to allow the incorporation in OPNET of software and libraries developed in MATLAB. Rafael Puerto [4]. presented architecture to perform real time executions. The application makes use of MATLAB Web Server capabilities, which enables to receive data from internet applications and enables to work with them in MATLAB. On the other hand, the distribution of user-defined Matlab toolboxes and rapid prototyping of many coarse-grained parallel applications had been discussed by Yuliya V. [5], the implementation was made available as a suite of three Matlab toolboxes.

The objective of this paper is to integrate C# and Matlab components in a grid network by using application program interface (API), .Net Remoting and the facilities provided by .Net framework 2.0, then investing this integration by building a client- administrator application that implements distributed computing .This is achieved by considering a case study showing the architecture of the application ICMCDC (Integrated C# and Matlab Components into Distributed Computers).

This paper organized as follows: An introduction to Grid computer and Cluster Computer has been presented in section (1). A background for the distributed processing and the technologies and standards used to construct distributed computations are presented in section (2). Section (3) described the architecture of the proposed application ICMCDC by presenting the design and implementation. The experimental results by using five computers as Clients and one computer as administrator are discusses in section (4). Finally, the paper is concluded and suggestions for future work were presented in section (5).

2. Distributed Processing [6].

The distributed processing solves complex problems that require a high computational calculation. These methods divide a big calculation to simpler and more independent problems, so that each simple problem can be solved by a different machine. Figure (1) shows a graphic representation of these concepts.

Distributed processing systems contain two differentiated elements: the central server, and the clients. The server divides the problem to simple problems and distributes these simple problems among the clients that will make the work. On the other hand, the server also has to join the results of all the

simple problems to obtain the wanted final solution. So that a task can benefit of the advantages of the distributed processing, this should be divisible in completely independent subtasks, so that these can be carried out simultaneously in several different machines. Of course, if a task needs to know the results of the previous task, the system could not execute them simultaneously.

2.1. Distributed Computing [7].

It's a method of computer processing in which different parts of a program are run simultaneously on two or more computers that are communicating with each other over a network. There are numerous technologies and standards used to construct distributed computations, including some which are specially designed and optimized for that purpose, such as:

- Remote Procedure Calls (RPC).
- Remote Method Invocation (RMI).
- . NET Remoting.

However, each of these technologies will be briefly described.

Remote Procedure Calls (RPC) [8].

RPC is a powerful technique for the development of client server distributed applications. It is based on extending the notion of conventional or local procedure calling, so that the called procedure need not exist in the same address space as the calling procedure. The two processes may be on the same system, or they may be on different systems with a network connecting them by using RPC.

Remote Method Invocation (**RMI) [9].**

Java-RMI enables the Java programmer to create distributed Java-to-Java applications, in which the methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts.

A Java program can make a call on a remote object once it obtains a reference to the remote object, either by looking up the remote object in the bootstrap naming service provided by RMI or by receiving the reference as an argument or a return value. A client can call a remote object in a server, and that server can also be a client of other remote objects.

RMI applications often use the client-server model. A typical server application creates a number of remote objects, makes references to those remote objects accessible, and waits for clients to invoke methods on those remote objects. A typical client application gets a remote reference to one or more remote objects in the server and then invokes methods on them. RMI provides the mechanism by which the server and the client communicate and pass information back and forth. Such an application is sometimes referred to as a distributed object application.

.NET Remoting.

It's a Microsoft application programming interface (API) for intercrosses communication released in 2002 with the 1.0 version of .NET Framework. It is one in a series of Microsoft technologies that began in 1990 with the first version of Object Linking and Embedding (OLE) for 16-bit Windows [11].

.NET Remoting was introduced in order to enable application components to interact easily across widely spread networks, it makes it possible for an application to use objects on the same computer or on any other computer that is

reachable across the network. Using .Net remoting to build an application in which two computers communicate directly across an application boundary, only the following need to be built:

- A remotable object.
- A host application domain to listen for requests for that object.
- A client application domain that makes requests for that object.

2.2. The Microsoft .NET Framework.

The .NET Framework is something like a (Java) virtual machine. It allows runtime environment functionality to any .NET application on whatever hardware platform or operating system, where the .NET Framework is implemented. The .NET is based on Common Language Infrastructure (CLI) technology, which ensures a right communication between independent application and specific hardware or operating system. CLI is used by many libraries, which are extending it. These libraries are referred to as frameworks. Actually, C# is really the native language of the .NET. However, .NET applications can be written in any language, which meets some requirements given by specifications (concretely the Common Language Specification). The language interoperability is conditioned by some mechanisms as Common data Types System (CTS), data marshalling, etc. All the .NET languages at the same way share the CTS [2].

2.3. Matlab Integration.

Matlab is powerful high-level language created by math works, and was developed to overcome the weaknesses that existed in competitive programming language like FORTRAN, C++ and C#. Matlab application contains all the necessary features of a complete development environment like coding, design, data manipulation, problem solving, graphic processing and a group of interactive tools. Matlab can be integrated with another application in three ways:

- ActiveX.
- Dynamic Data Exchange (DDE).
- Application Program Interface (API).

ActiveX is a protocol included in Microsoft windows consists of a number of object oriented technologies that have common root called COM (Component Object Module). Matlab on the other hand can deal with two ActiveX technologies which are ActiveX controls and ActiveX automation. The former are application components that can be visually and programmatically integrated while the latter enables Matlab to control or be controlled by other ActiveX components.

DDE is an old and powerful service of windows that allow the application to communicate and exchange data. The two parameters needed to be defined during a DDE conversation are the first is the name of the application providing the service called service name, and the second is the conversation subject which is called the topic.

So, a conversation is defined by a service and topic when a request for a service is received concerning a precise topic, the server returns an acknowledgment and a DDE conversation is established. The data exchanged between the client and server application during a DDE conversation is called an Item, which is a reference to data that is

meaningful to client and server applications in a conversation .as shown in figure (3).

API is used to interact Matlab with external data and programs, therefore we can say that API is Malta's interface to the out side world. API functions include the following:

1. Calling C#, C, or other programs from MATLAB.
2. Importing and exporting data to and from the MATLAB environment.
3. Creation client/server relationships between MATLAB and other Software programs.

3. ICMCDC application.

A software application enables user to access each client remotely from the administrator using Microsoft .Net remoting has been built. The software called ICMCDC (Integrated C# and Matlab Component into Distributed Computer) application. ICMCDC was built by taking advantage of the facility provided by Microsoft .net framework, ICMCDC is a server-client application using C# integrated with Matlab component is. Also, DirectX is used to accomplish the job execution example presented in the ICMCDC application for graphic display rendering representation.

The architecture of ICMCDC application has been illustrated in section (3.1) to show the administrator /client relationship and how the jobs are distributed to the clients through a computer grid, executed in the clients and then returning the results to the administrator computer. It also illustrates how this is achieved by building C# based administrator and client software application integrated with Matlab which was used to calculate the results.

3.1. ICMCDC Architecture.

This architecture describes ICMCDC application and the flow of integration between the client and administrator. Figure (4) shows the ICMCDC application control flow.

ICMCDC is presented as single install package which is installed in the same manner on both administrator and client. The following steps show the Architecture for the application:

1. ICMCDC is installed on both administrator and client computers in the same manner; the administrator or client mode selection is achieved through the main ICMCDC window.
2. On the client side, when pressing on the do job, the administrator will send a remote instruction to the clients to start executing the jobs assigned to them.
3. On the client side, the client will call Matlab to execute the job and then return the results to the administrator.

As a result, the administrator will collect the result delivered by each client and display these results. The detail of ICMCDC architecture is shown in Figure (5).

3.2. Formulating ICMCDC.

The ICMCDC interfaces were built using Microsoft visual studio. By selecting new project from the file menu, a new dialog is presented containing options for language programming of the project which can be C#, java, or C++. After selection, the type of templates that will be used must be chosen, then window application, and then a name for the project is given as shown in figure (6).

The start will be to build the client and administrator interfaces through programming window provided by Microsoft visual studio which contains the toolboxes and common controls required to build the client and administrator interface as shown in figure (7).

When adding a reference from the .Net framework library is needed, we have to go to the project tab and then add new reference so a window appears containing all the references provided by .Net framework, so the needed reference can be selected and press OK. By that all the classes contained under this reference will be available as shown in figure (8).

3.3. ICMCDC implementation.

In General, C# and Matlab integration can be of great benefit when using grid computing. That is because C# is very powerful in establishing the grid connection, data exchange, data marshalling and user interfaces, while Matlab is powerful in the mathematical and functional calculations therefore we gain the power of both C# and Matlab through this integration to achieve an ideal grid computing environment and reach the goal expected from the computer grid which is load balancing throughout all the peers of the grid. Load balance is the processes of distributing load equally to multi executors which are distribute equal jobs to different computers. This process by no doubt provides time reduction since several jobs will be executed at the same time.

Figure (9) shows a case study for ICMCDC Application Implementation. In this case study, a single administrator and five clients were considered and connected via a network. ICMCDC is installed on all the six computers in the same manner, graphic and render jobs were created and executed on the network as follows;

ICMCDC implementation was taken from a five-node LAN connected as 100 Base TX fast Ethernet via 24 port 10/100 mbps duel speed stackable switch. The specification of the computers that function as administrator and clients of the ICMCDC application are:

- CPU: Pentium4 with 2.6 GHZ.
- RAM: 1 GByte.
- Operating System: Microsoft windows XP professional for all PCs (Administrator (station), clients (workstations)).
- Microsoft .net framework 2.0: that helps building client-administrator application.
- DirectX 0.9: that helps in rendering the graphics that we use as jobs and are sent from client to be displayed on the administrator.
- Matlab 0.7 that is used to make the computations for doing the job.

3.4. The Administrator

The administrator application window provides two facilities, the first is to give the "do job in client" order from the administrator window which forces job execution on all clients that are connected to the administrator at one time which is in fact the principle of remote execution. The second facility is to give the "do job" order from the client itself, and in this case the order must be done on every client connected to the administrator independently.

Note that before any of the two methods mentioned previously is

implemented, the execution environment must call on the administrator through the "do job in Administrator" button as shown in figure (10).

3.5. The Clients

The client application window contains the "do job" button which enables the client to control the job execution if required, or it could receive a do job command remotely from the administrator PC, this means that we have the option to give the do job command either from the administrator PC or from the client PC as shown in figure (11).

4. The results in the Administrator

In the administrator computer when pressing on the do job button, the environment will be displayed as shown in figure (12)**,** it is blank waiting for results from clients

When the first client connects to the administrator PC and presses on the do job button then the job will be executed and the results are sent to the administrators that are displayed in the window as shown in figure (13)**,** it is first tree from the client (1).

By the same way the other four clients will send the results of their different trees for the administrator. Figure (14) displays the final output, when all the five clients send their results and displayed in the administrator PC.

It is valuable to notice that when we press on do job in client button any client that is connected to the administrator PC will send its results.

5. Conclusions and suggestions for future work

In spite of Matlab is a self contained high level computational language it lacks user friendly interfaces, which might limit gaining the full benefit of Matlab functionalities, therefore C# language has completed this shortage to reach the optimum performance. Also, Matlab grid has added user supportive capability to Matlab, most important is the ability to execute very huge jobs which extend any existing computer processor limitations, making Matlab grid a very powerful and a useful tool.

Adding backup servers to the master server in the ICMCDC may be a further work for this paper. The backup Server could be contacted only in the case the main server is not responding to the client's request, thus preventing any downtime of the MatlabGrid. Additionally, supporting the wireless capability to provide a wireless distribution of jobs and functions through wireless ports which are considered a new generation of communications and networking, taking into account the security issue which arises with this implementation.

References

[1].Anand Natrajan, Michael P. Walker,"Monitoring Remote Jobs in a Grid System", @virginia.edu, 2000.

[2].Anne E. Trefethen, "MultiMATLAB: MATLAB on Multiple Processors", 1996.

[3].Vikram Dham, "Link Establishment in Ad Hoc Networks Using Smart Antennas", MSc. thesis, Faculty of the Virginia Polytechnic Institute and State University, 2003.

[4].Rafael Puerto, "Remote Control Laboratory Using Matlab and

Simulink: Application to ADC Motor Model", 2004.

[5].Yuliya V. Karpievitch and Jonas S. Almeida,"mGrid: A Load-Balanced Distributed Computing Environment for the Remote Execution of the User-Defined Matlab Code", BioMed Central Ltd, 2006.

[6].Gabrielle Allen," The Grid Application Toolkit: Towards Generic and Easy Application", proceedings of the IEEE, 2004.

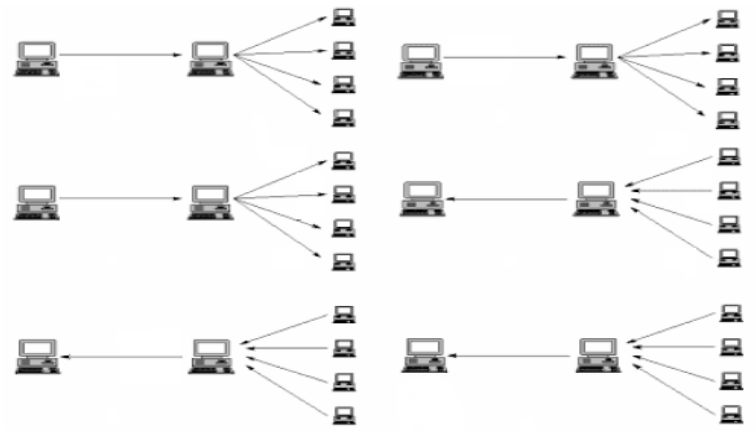
[7].Parallab, "Software Reusability and Efficiency", Bergen Center for Computational Science, University of Bergen, 2004.

[8].Jonas Borgström," ARIA and Matlab Integration With Applications ", MSc. thesis, Umea University, 2005.

[9]."Grid and Cluster Computing: Options for Improving Windows Application Performance ", Digipede Technologies, LLC, 2003.

[10].David Shank, "Office VBA and the Windows API ", MSDN magazine, March 2001.

[11].Cleve Moler,"Why there isn't a parallel MATLAB", MathWorks Newsletter. Spring, 1995.



(a) (b)

Figure (1) Distributed processing example [6]. (a) Server sends the job as two consecutive bursts for the clients. (b) Clients return the results of the job as two consecutive bursts for the server.

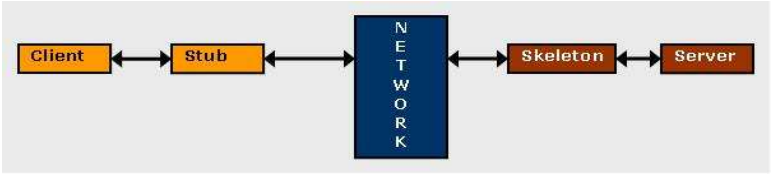


Figure (2) Implementation model of Java-RMI [10].

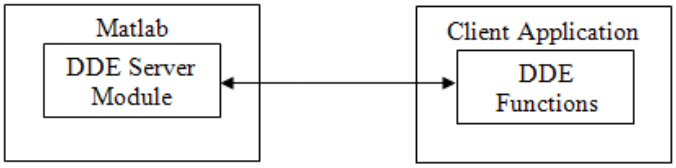


Figure (3) Conversation between DDE Client/Server.

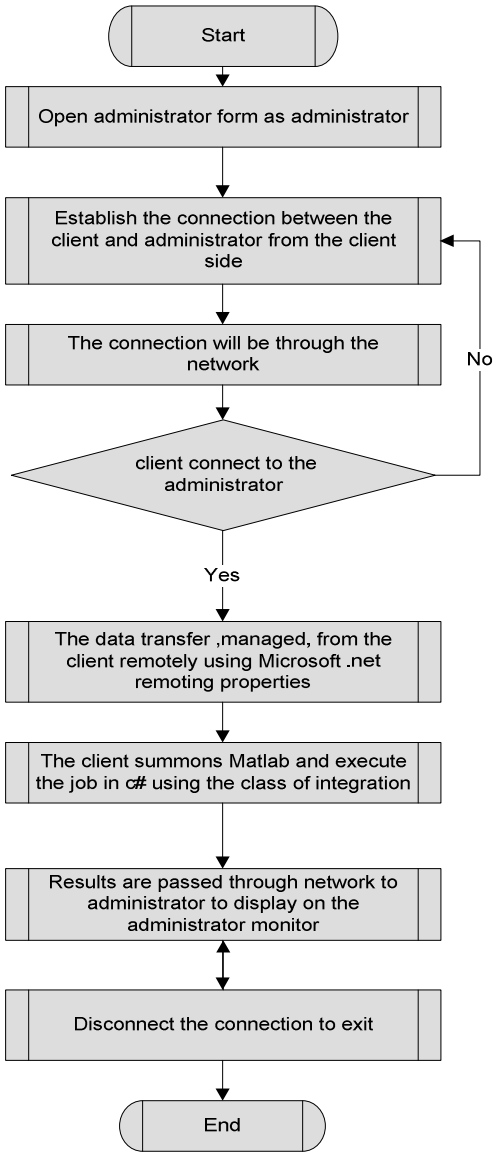
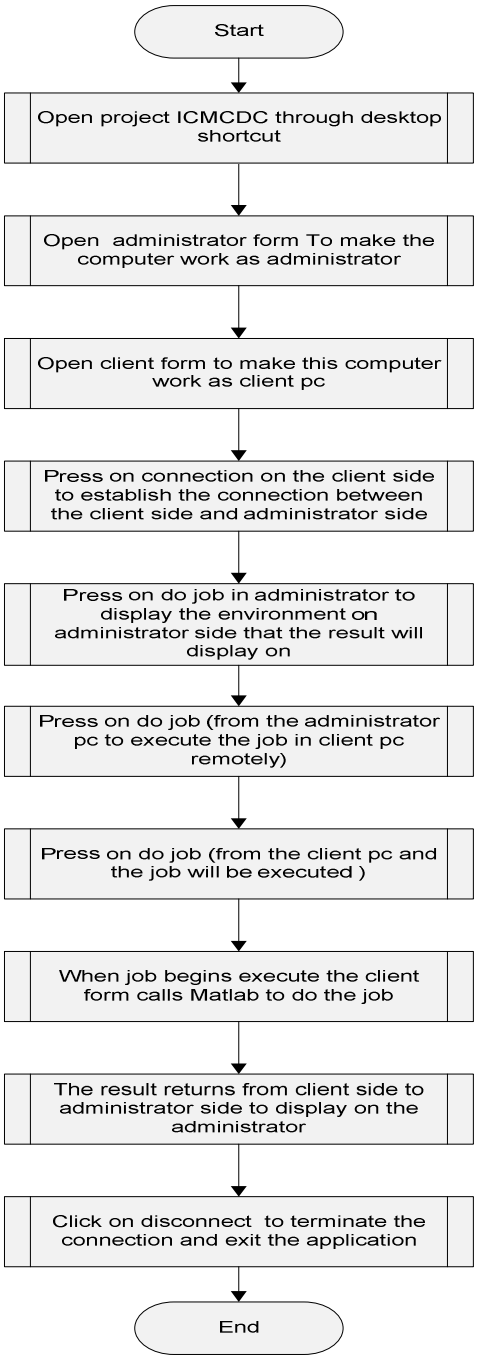


Figure (4) Control flow of ICMCDC Application.



Figur(5) *Computer Engineering and Information Technology Dept./University of Technology/Baghdad e ICMCDC Architecture.

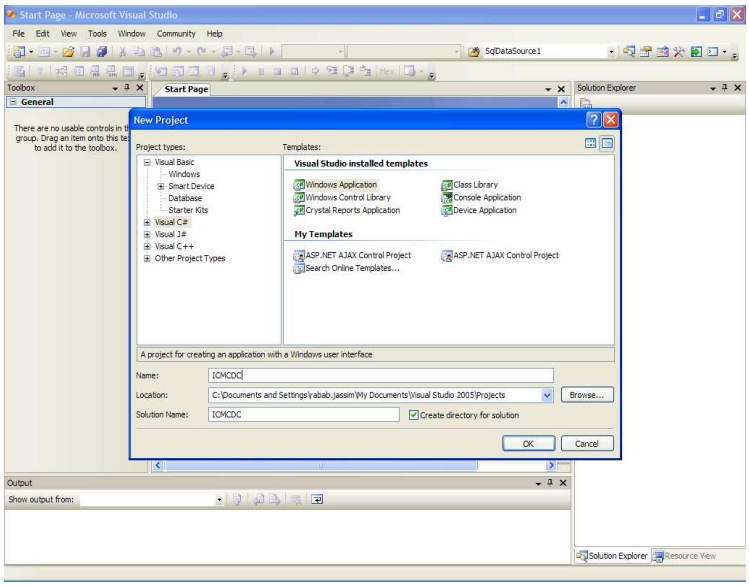


Figure (6) Visual studio project creation interface.

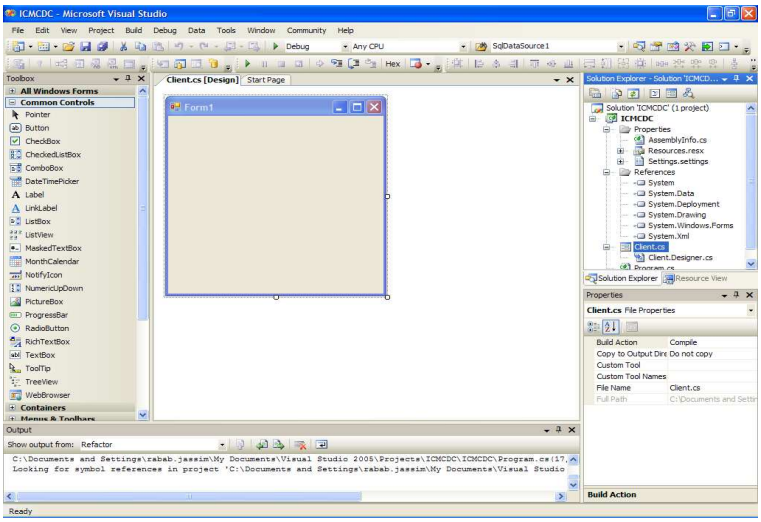


Figure (7) The environment for building the application.

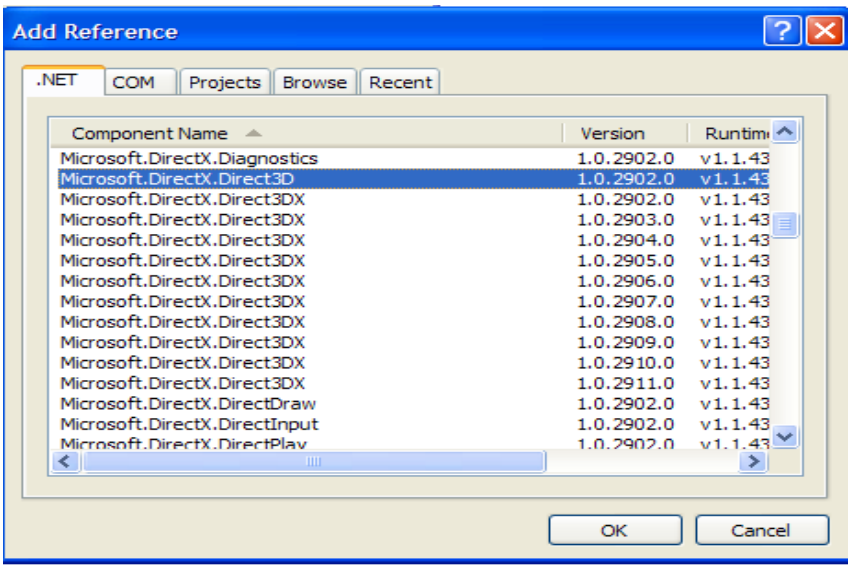


Figure (8) The invironment to add a reference from the .Net framework.

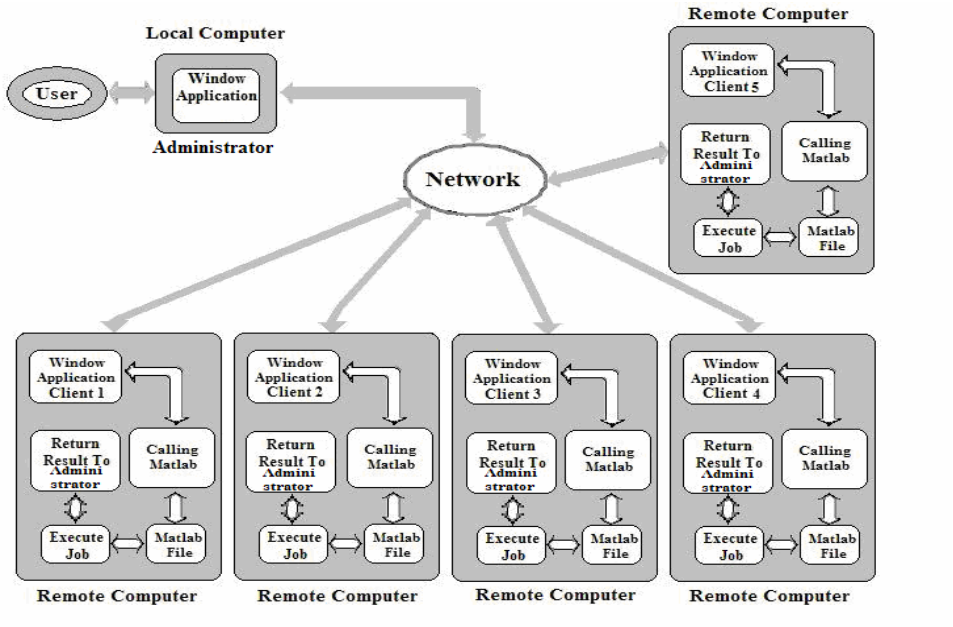


Figure (9) ICMCDC application architecture.

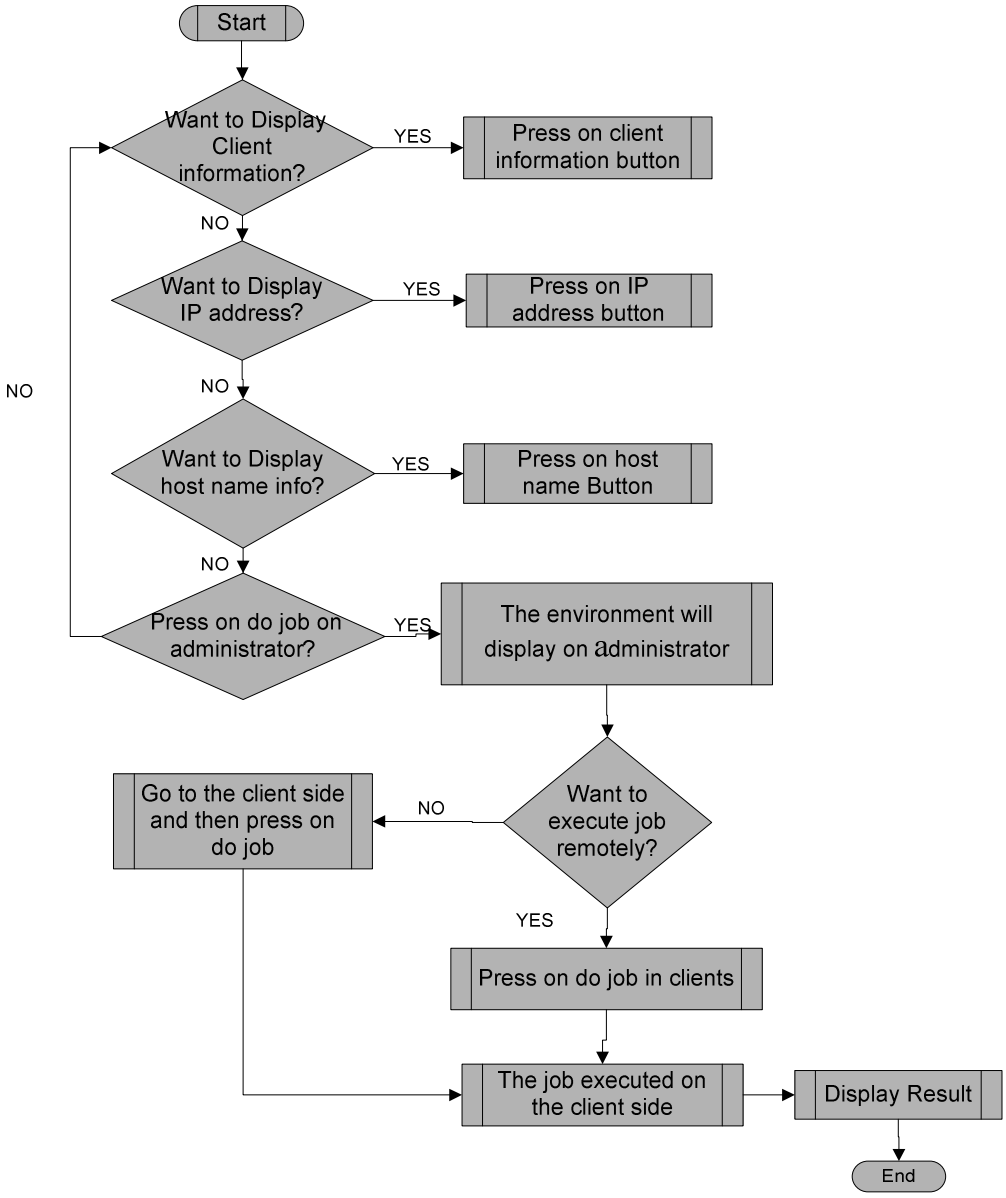


Figure (10) Flowchart of the Administrator.

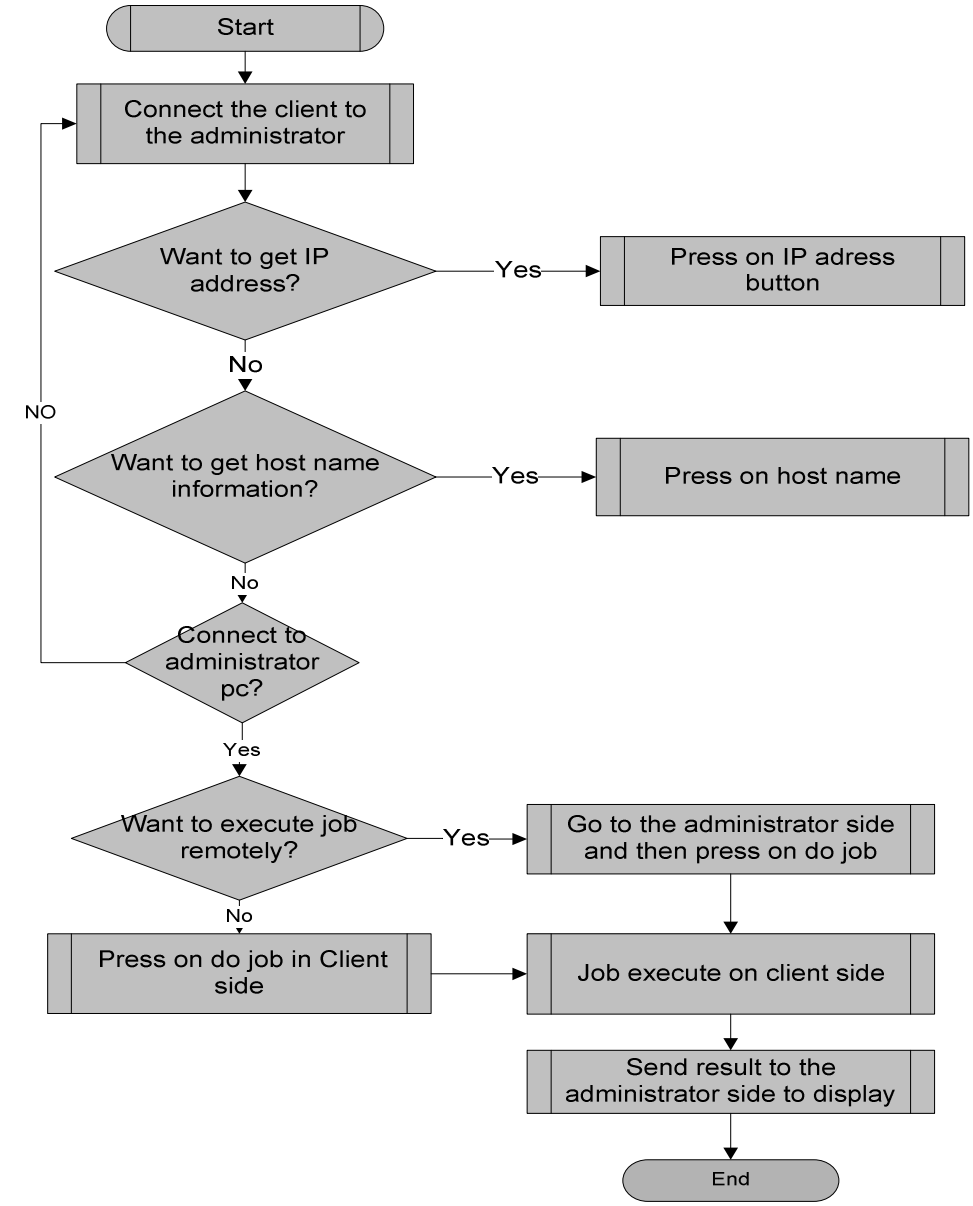


Figure (11) Flowchart of the client.

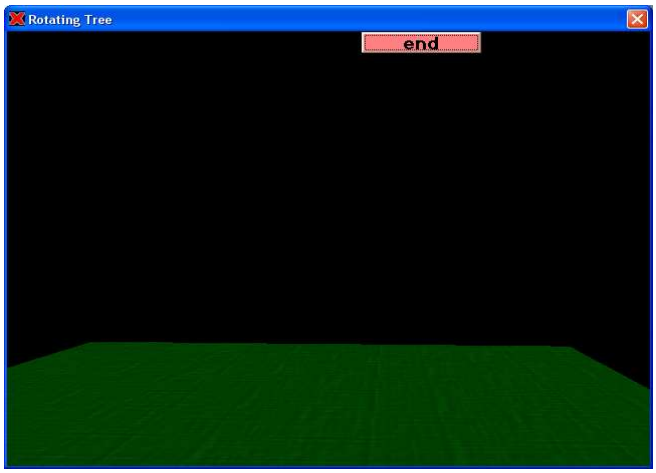


Figure (12) The window for the environment that will be used to display the results from clients

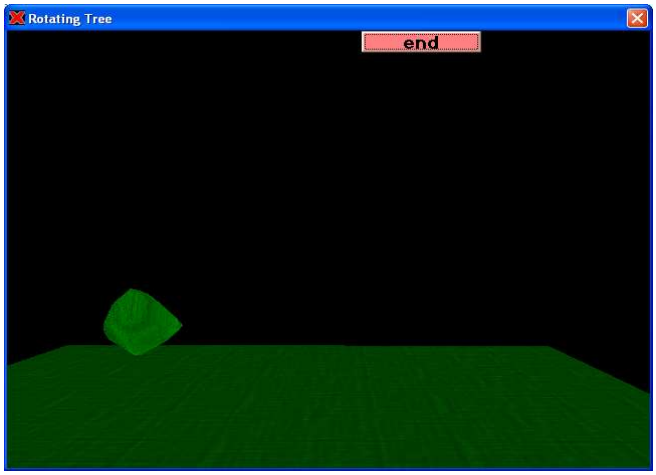


Figure (13) The window to show the job of client 1.

Figure (14) The result of five client jobs as shown in the administrator.